

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellant : Taher ELGAMAL et al.                      Art Unit : 2135  
Serial No. : 09/920,801                                      Examiner : Klimach, Paula W.  
Filed : August 3, 2001  
Title : CRYPTOGRAPHIC POLICY FILTERS AND POLICY CONTROL METHOD  
AND APPARATUS

**Mail Stop Appeal Brief - Patents**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

BRIEF ON APPEAL

**(1) Real Party in Interest**

AOL, LLC., the assignee of this application, is the real party in interest.

**(2) Related Appeals and Interferences**

There are no related appeals or interferences.

**(3) Status of Claims**

Claims 31-34, 36-43 and 45-48 are now pending, of which claims 31 and 40 are independent. All claims have been rejected, and all claims have been appealed.

**(4) Status of Amendments**

The claims have not been amended subsequent to the final rejection.

**(5) Summary of Claimed Subject Matter**

Independent claim 31 is directed to a method for controlling functions of an application program. The method includes accessing a policy file that includes an attribute portion configured to store one or more policy attributes and a value portion having one or more attribute values.<sup>1</sup> Each attribute value corresponds to a policy attribute and indicates whether an application program may access the function represented by the policy attribute.<sup>2</sup> Each policy

---

<sup>1</sup> See e.g., specification, page 10, lines 1-18

<sup>2</sup> See e.g., specification, page 10, table 1

file includes a signature portion with at least one digital certificate.<sup>3</sup> The method includes determining whether the policy file is unaltered based on the signature portion of the policy file.<sup>4</sup> The method also includes retrieving at least one of the attributes, and, for each retrieved attribute, an attribute value corresponding to the attribute from the policy file.<sup>5</sup> The method further includes determining whether a function represented by a retrieved attribute is permitted to be accessed by the application program and permitting the application program to access the function conditioned upon a determination that the policy file is unaltered.<sup>6</sup>

Independent claim 40 is directed toward an apparatus for controlling functions of an application program. The apparatus is configured to, access a policy file that includes an attribute portion configured to store one or more policy attributes and a value portion having one or more attribute values,<sup>7</sup> each attribute value corresponding to a policy attribute and indicating whether an application program may use a function capable of being performed by the application program and represented by the policy attribute,<sup>8</sup> and a signature portion including at least one digital certificate.<sup>9</sup> The apparatus also is configured to determine whether the policy file is unaltered based on the signature portion of the policy file<sup>10</sup> and retrieve at least one of the attributes and, for each retrieved attribute, an attribute value corresponding to the attribute from the policy file.<sup>11</sup> The apparatus is further configured for determining whether a function represented by a retrieved attribute is permitted to be accessed by the application program, and for permitting the application program to access the function conditioned upon a determination that the policy file is unaltered.<sup>12</sup>

---

<sup>3</sup> See e.g., specification, page 12, lines 5-8

<sup>4</sup> See e.g., specification, page 12, lines 8-12

<sup>5</sup> See e.g., specification, page 11, table 2

<sup>6</sup> See e.g., specification, page 14, lines 22-34

<sup>7</sup> See e.g., specification, page 10, lines 1-18

<sup>8</sup> See e.g., specification, page 10, table 1

<sup>9</sup> See e.g., specification, page 12, lines 5-8

<sup>10</sup> See e.g., specification, page 12, lines 8-12

<sup>11</sup> See e.g., specification, page 11, table 2

<sup>12</sup> See e.g., specification, page 14, lines 22-34

**(6) Grounds of Rejection**

**a. Claims 31, 36-40, and 45-48 under 35 U.S.C. § 103**

Claims 31, 36-40, and 45-48 under 35 U.S.C. § 103 as being unpatentable over Drews (U.S. Patent No. 6,647,494) in view of Perona (U.S. Patent No. 6,671,809).

**b. Claims 32-34 and 41-43 under 35 U.S.C. § 103**

Claims 32-34 and 41-43 also were rejected under 35 U.S.C. § 103 as being unpatentable over Drews in view of Perona, further in view of Anderl (WO 87/07063).

**(7) Argument**

**a. Claims 31, 36-40, and 45-48 under 35 U.S.C. § 103 are not properly rejected under 35 U.S.C. § 103 as being unpatentable over Drews in view of Perona.**

Appellant requests reversal of this rejection because Drews in view of Perona does not teach or suggest the subject matter of the independent claims 31 and 40, as described more fully below. In particular, Drews, Perona, and the proposed combination each fail to teach or suggest the claimed limitation and requirement for “each attribute value corresponding to a policy attribute and indicating whether an application program may use a function capable of being performed by the application program and represented by the policy attribute.”

**Independent Claim 31 and Dependent Claims 36-39**

Independent claim 31 is directed toward a method for controlling functions of an application program. The method includes, *inter alia*, accessing a policy file that includes an attribute portion configured to store one or more policy attributes and a value portion having one or more attribute values, each attribute value corresponding to a policy attribute and indicating whether an application program may use a function capable of being performed by the application program and represented by the policy attribute, and a signature portion including at least one digital certificate. The method also includes, *inter alia*, determining whether the policy file is unaltered based on the signature portion of the policy file and retrieving at least one of the attributes and, for each retrieved attribute, an attribute value corresponding to the attribute from

the policy file. The method further includes, *inter alia*, determining whether a function represented by a retrieved attribute is permitted to be accessed by the application program and permitting the application program to access the function conditioned upon a determination that the policy file is unaltered.

Appellant requests reversal of the rejection to claim 31, as a consequence of neither Drews, Perona, nor the combination of the proposed references teaching or suggesting the use of attribute values, each of which corresponding to a policy attribute and indicating whether an application program may use a function capable of being performed by the application program and represented by the policy attribute, as recited by claim 31. Below, Appellant provides an explanation of why Drews, Perona and their proposed combination fails, and thereafter, Appellant explains why this failure of Drews and Perona is not cured by comments offered by the Examiner in the Advisory Action.

*Drews does not teach or suggest the claimed attribute values*

Drews discloses a method and system for checking authorization for program additions or updates. In order to ensure authorization of the delivered program, a client sends authorization information to the host, which uses the authorization information to generate and deliver the program additions or updates to the client.<sup>13</sup> Drews refers to the authorization information as an “update token,” and the generated and delivered program addition or update as a “credential manifest.”<sup>14</sup>

Among other aspects of Drew, the Final Office Action references its disclosure of the “manifests.” A manifest (i.e. “request credential manifest”) includes an update token, a list of configurable parameter to be updated, a list of new values for those configurable parameters, and a manifest digital signature. The “update token is a hash value that the console platform needs [in order] to construct a request credential manifest.” As for the other components of Drew’s manifest, the list of configurable parameters is the software to be updated or added and the manifest digital signature is used for verification. None are attributes that indicate whether an application program may use a function.

---

<sup>13</sup> See Drews, FIG. 3

<sup>14</sup> See Drews, column 3, line 3 to column 4, line 17

Stated differently, the claimed policy file requires attribute values which indicate “whether an application program may use a function...represented by the policy file,” and the manifest of Drews simply does not include such attribute values, nor does the Final Office Action contend so. In fact, to the contrary, the Final Office Action acknowledges the failure of Drews to include attributes revealing whether an application can use a feature, stating:

**Drews does not disclose expressly disclose [sic] determining whether an application program may use a function capable of being performed by the application program and thus determining whether a function represented by a received attribute is permitted to be accessed by the application program; and permitting the application program to access the function conditioned upon the determination that the policy file is unaltered. [See page 3]**

Appellants acknowledge these shortcomings of Drews with particular reference to the claim limitation of “each attribute value corresponding to a policy attribute and indicating whether an application program may use a function capable of being performed by the application program and represented by the policy attribute.”

*Perona does not teach or suggest the claimed attribute values*

The Final Office Action relies on Perona to reject the remainder of the claim language:

**Perona discloses determining whether an application program may use a function capable of being performed by the application program and thus determining whether a function represented by a retrieved attribute is permitted to be accessed by the application program (column 6 lines 15-23)**

Like Drews, Perona fails to teach or suggest the claim limitation concerning attributes, namely the requirement for “each attribute value corresponding to a policy attribute and indicating whether an application program may use a function capable of being performed by the application program and represented by the policy attribute.”

Perona generally relates to open architecture software that enables software to be loaded on a platform, such as a cellular phone, by performing checks among system components based on configuration and rule information.<sup>15</sup> In Perona, software modules may be installed to update or add to an application on a platform. The software modules include rules with requirements that “must be met by the modules” for the platform to be able to load the modules.<sup>16</sup> In particular, the software modules include pointer record rules specifying what are the specific

<sup>15</sup> See Perona, column 1, lines 7-13 and FIG. 1

<sup>16</sup> See Perona, column 4, lines 14-16

requirements and limitations imposed on a given module.<sup>17</sup> Notably, in Perona, the rules are read, and based on the rules, a set of requirements or limitations are assigned, as necessary, for installation to take place. If the device is able to meet the requirements or limitations assigned by the rules, the platform loads the module for execution.<sup>18</sup>

While Perona's rules enable a process to determine what requirements or limitations are needed to be met in order to determine whether or not a target computer can load or use an application, nowhere does Perona teach or suggest using attributes values which specifically indicate (or used in a look-up table to indicate) whether an application program may use a function capable of being performed by the application program and represented by the policy attribute.

More specifically, the cited portion of Perona describes rules that are used to construct a process to assign module requirements or constraints that must be met before the module can be loaded onto the platform. Notably, while the claim limitation requires the use of attribute values indicating whether an application program may use a function, Perona uses rules to create a process to determine whether application specific requirements are met. Specifically, the rules of Perona enable the device to develop a process to determine whether or not factors *external to the rules* permit the application to be loaded. For example, Perona details a wireless information transmitting system where the rules (i.e., module requirement) specify a waveform requirement of an RF module. "Before the platform 20 can load the referenced module, the platform must verify the limitations and requirements specified in the module rules record 46 are met."<sup>19</sup> In contrast, the claimed attribute values themselves "[indicate] whether an application program may use a function."

This distinction is not trivial. Rather, it is quite important, as it allows the claimed invention to achieve results not attainable by Perona. By way of example, the specification details an implementation in which software is loaded with multiple available functions, and with the determination of which functions are permitted stored in the attribute values.<sup>20</sup> In this manner, the determination of whether the functions are permitted is made prior to the loading of

---

<sup>17</sup> See Perona, column 4, lines 26-32

<sup>18</sup> See Perona, column 4, lines 21-43

<sup>19</sup> See Perona, column 4, lines 37-43

<sup>20</sup> See e.g., FIG. 2

the software on the device, and prevent may prevent some possible tampering or tricking the software.

Therefore, claim 31 defines an invention that is patentable over Drews in view of Perona, as do pending dependent claims 36-39. Accordingly, Appellants requests reconsideration and withdrawal of the imposed rejection.

As a more specific example of the differences between Perona's use of rules and module requirements and the claimed attribute values, dependent claim 38 describes one implementation in which a truth expression is included in each of the attribute values where the truth expression is one of a true flag, a false flag, and a conditional flag which may enable, disable, or conditionally enable an available function of a program.

*Advisory Action's non-specific and conflicting response*

Appellant's response to the Final Office Action pointed out that the Final Office Action did not specifically address the claim limitation of each attribute value indicating whether an application program may use a function capable of being performed by the application program and represented by the policy attribute. The Advisory Action responded by stating (emphasis added):

**Drews discloses each attribute value corresponding to a policy attribute and indicating whether an application program may use a function capable of being performed by the application program (column 4 line 56 to column 5 line 45).**

Appellant respectfully disagrees with the Advisory Action and maintains that Drews fails to disclose each attribute value indicating whether an application program may use a function capable of being performed by the application program and represented by the policy attribute.

Notably, this aspect of Drews is identified for the first time in this Advisory Action. Also of note is that the above citation (column 4 line 56 to column 5 line 45) spans 6 paragraphs and a discussion of 4 figures of Drews. In order to facilitate understanding by the Appellant (and the Board) of which portion of Drews is now believed by the Examiner to support this position, it would be helpful to receive a more concise identification from the Examiner of the particular language within Drews that is believed to disclose the subject limitation.

Although not specifically stated, Appellant assumes that the Advisory Action maintains that the attribute values are disclosed by the manifest in Drews at column 4, lines 17-21 as the manifest was mentioned in the Final Office Action, even though the Examiner seems to appreciate the failure of the manifest to address the rest of the limitations of the claimed attribute values.<sup>21</sup> As discussed above, the manifest in Drews does not teach or suggest each attribute value indicating whether an application program may use a function capable of being performed by the application program and represented by the policy attribute.

Returning to the Advisory Action:

**The appellant argues that Perona does not disclose or suggest each attribute value corresponding to a policy attribute and indicating whether an application program may use a function capable of being performed by the application. Although this is true...Perona discloses attribute values that indicate whether an application program may use a function capable of being performed by the application program (column 6 lines 15-23)**

Perona also does not teach or suggest attribute values that indicate whether an application program may use a function capable of being performed by the application program and represented by the policy attribute. Specifically, Perona, at the portion relied on in the Advisory Action above, states:

**Also, the platform 20 at 66 accesses and checks the integrity of the module 24 by checking the module identification record 50. Subsequently, the platform 20 at 68 checks the integrity of both the application 22 per the application identification record 32 and itself at 70 per the platform rules and configuration information against the module rules record 52 to determine if both the application 22 and the platform 20 meet all requirements of the module 24**

In Perona, the platform 20, the application 22, the module identification record 50, and the module rules record 52 are all parts of a block diagram of system components.<sup>22</sup> The cited portion is seen to show parts and checks of a software system that are used to decide whether to load modules based on requirements/constraints. Alternatively, the claim limitation includes accessing a policy file that includes attribute values indicating whether an application program may use a function capable of being performed by the application program.

**Independent Claim 40 Dependent Claims 45-48**

---

<sup>21</sup> See Final Office Action, page 2, last paragraph

<sup>22</sup> See Perona, column 2, lines 44-51



Similar to claim 31, independent claim 40 recites a policy file that includes attribute values, each attribute value corresponding to a policy attribute and indicating whether an application program may use a function capable of being performed by the application program and represented by the policy attribute. For the reasons above with respect to claim 31, appellants submit that the rejection of independent claim 40 and dependent claim 45-48 should be withdrawn.

**b. Claims 32-34 and 41-43 are not properly rejected under 35 U.S.C. § 103(a) as being unpatentable Drows in view of Perona, further in view of Anderl.**

As claims 32-34 and 41-43 are all dependent on independent claim 31 or independent claim 40, Appellant requests reversal of this rejection because neither Drows, Perona, Anderl, nor any proper combination of the references teaches or suggests the subject matter of the independent claims 31 and 40. For example, neither Drows, Perona, or Anderl teach or suggest "each attribute value corresponding to a policy attribute and indicating whether an application program may use a function capable of being performed by the application program and represented by the policy attribute and represented by the policy attribute." Specifically, Anderl does not make up for the deficiencies of Drows in view of Perona, nor does the Final Office Action contend so.

For at least these reasons, appellant requests reversal of the § 103 rejection of claims 32-34 and 41-43.

**c. Conclusion**

For the foregoing reasons, the rejections should be reversed.

In accordance with appellant's Notice of Appeal filed June 06, 2006, appellant submits this Appeal Brief.

Appellant : Taher ELGAMAL et al.  
Serial No. : 09/920,801  
Filed : August 3, 2001  
Page : 10 of 15

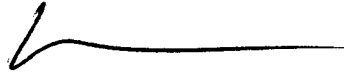
Attorney's Docket No.: 06975-193002 / Security 20-  
CON

The fee for the appeal in the amount of \$500 and the fee for a Four-Month Extension of Time in the amount of \$1590 is being paid concurrently herewith on the Electronic Filing System (EFS) by way of Deposit Account authorization.

Please apply any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: 10/10/2006

  
\_\_\_\_\_  
W. Karl Renner  
Reg. No. 41,265

Fish & Richardson P.C.  
1425 K Street, N.W.  
11th Floor  
Washington, DC 20005-3500  
Telephone: (202) 783-5070  
Facsimile: (202) 783-2331

### **Appendix of Claims**

31. A method for controlling functions of an application program, the method comprising:

accessing a policy file that includes an attribute portion configured to store one or more policy attributes and a value portion having one or more attribute values, each attribute value corresponding to a policy attribute and indicating whether an application program may use a function capable of being performed by the application program and represented by the policy attribute, and a signature portion including at least one digital certificate;

determining whether the policy file is unaltered based on the signature portion of the policy file;

retrieving at least one of the attributes and, for each retrieved attribute, an attribute value corresponding to the attribute from the policy file;

determining whether a function represented by a retrieved attribute is permitted to be accessed by the application program;

permitting the application program to access the function conditioned upon a determination that the policy file is unaltered.

32. The method of claim 31 wherein the policy file comprises a JAVA archive file.

33. The method of claim 31 wherein the policy file comprises multiple component files, at least one of the component files storing some of the attribute portions and attribute values.

34. The method of claim 33 wherein at least one of the multiple component files is associated with a signature portion including at least one digital certificate for ensuring that the policy file has not been modified and a signature portion including at least one digital certificate for ensuring that the policy file has not been modified and applying to a particular component file.

35. (Cancelled)

36. The method of claim 31 wherein:  
the signature portion applies to the attribution portion and the value portion of the policy file;

determining whether the policy file is unaltered comprises determining whether the attribute portion and the value portion are unaltered based on the signature portion.

37. The method of claim 36 wherein the signature portion applies to the policy file.

38. The method of claim 31 wherein:  
each of the attribute values is one of a string, an integer number, and a truth expression.

39. The method of claim 38 wherein the truth expression is one of a true flag, a false flag, and a conditional flag.

40. An apparatus for controlling functions of an application program, the apparatus being configured to:

access a policy file that includes an attribute portion configured to store one or more policy attributes and a value portion having one or more attribute values, each attribute value corresponding to a policy attribute and indicating whether an application program may use a function capable of being performed by the application program and represented by the policy attribute, and a signature portion including at least one digital certificate;

determine whether the policy file is unaltered based on the signature portion of the policy file;

retrieve at least one of the attributes and, for each retrieved attribute, an attribute value corresponding to the attribute from the policy file;

determining whether a function represented by a retrieved attribute is permitted to be accessed by the application program; and

permitting the application program to access the function conditioned upon a determination that the policy file is unaltered.

41. The apparatus of claim 40 wherein the policy file comprises a JAVA archive file.
42. The apparatus of claim 40 wherein the policy file comprises multiple component files, at least one of the component files storing some of the attribute portions and attribute values.
43. The apparatus of claim 42 wherein at least one of the multiple component files is associated with a signature portion including at least one digital certificate for ensuring that the policy file has not been modified and the signature portion applying to a particular component file.
44. (Cancelled)
45. The apparatus of claim 40 wherein the signature portion applies to the attribute portion and the value portion of the policy file;  
determining whether the policy file is unaltered comprises determining whether the attribute portion and the value portion are unaltered based on the signature portion.
46. The apparatus of claim 45 wherein the signature portion applies to the policy file.
47. The apparatus of claim 40 wherein:  
each of the attribute values is one of a string, an integer number, and a truth expression.
48. The apparatus of claim 47 wherein the truth expression is one of a true flag, a false flag, and a conditional flag.

Appellant : Taher ELGAMAL et al.  
Serial No. : 09/920,801  
Filed : August 3, 2001  
Page : 14 of 15

Attorney's Docket No.: 06975-193002 / Security 20-  
CON

### **Evidence Appendix**

None

Appellant : Taher ELGAMAL et al.  
Serial No. : 09/920,801  
Filed : August 3, 2001  
Page : 15 of 15

Attorney's Docket No.: 06975-193002 / Security 20-  
CON

### **Related Proceedings Appendix**

None